A Motion Planning Method Based on Genetic Algorithms and Curve Fitting Techniques

Brock Kopp
Mechatronics Engineering
University of Waterloo
Waterloo, Ontario
brock.kopp@uwaterloo.ca

Karl Price
Mechatronics Engineering
University of Waterloo
Waterloo, Ontario
karldprice@gmail.com

Angelica Ruszkowski
Mechatronics Engineering
University of Waterloo
Waterloo, Ontario
aruszkow@uwaterloo.ca

***Abstract*—Navigation and path planning are critical components of developing robotic technology, where optimized path planning can improve efficiency and throughput of an industrial robotic manipulator. A genetic algorithm is used to plan a polynomial based trajectory which avoids collisions with any obstacles and minimizes trajectory path distance and dynamic constraints. The heuristic nature of the genetic algorithm allows it to determine a viable and efficient path for a manipulator quicky as it's environment changes. Algorithm results are verified by comparing with deterministic algorithms which produce the absolute best path.**

***Index Terms*—Genertic Algorithms, Curve Fitting, Path Planning, Grid Method, Robotics**

I. Introduction

Path planning is a critical component of navigation technology, a fundamental area in robotic research. Verifying path safety is critical for both the safety of the robot as well as its surrounding environment, which can contain people or valuable infrastructure. The path must ultimately reach its destination while satisfying specified criteria, the most common of which is distance (or time). Another important criteria is path dynamics, where the forces and other dynamic characteristics of the robot are considered. Excessive acceleration and jerk cause stress on the robotic manipulator as well as robot tracking characteristics which are often a function of acceleration. These criteria accounted for by relatively few common path planning algorithms [1].

Path planning techniques can be applied to industrial robotic manipulators to ensure efficient operation while maneuvering safely in the environment. Articulated manipulators are bounded to their configuration workspace, which is a two-dimensional representation of their allowable motion range [2]. Frequently used algorithms for motion planning include probabilistic roadmaps, potential field methods, and neural network approaches [3]–[6]. Since every robot has a unique and complex task and environment, a robust solution is necessary. Genetic algorithms are especially suited for such complex optimization problems [7].

As a stochastic optimization technique, a genetic algorithm will find the global optimum for a problem in a reasonable amount of time and computational cost. Due to the iterative evolutionary nature of genetic algorithms, computational time may be greater than other applicable algorithms. However, the robustness of a genetic algorithm, combined with its ability to easily evaluate all path criteria of the optimization problem make it an appealing solution.

The robot's environment and obstacles will be represented in a discretized grid, and polynomial curve fitting techniques will be used to define the optimum path from the start to the end location through the specified environment. The curve fitting method helps simplify and expedite the generation of a desired path, compared to commonly used techniques such as rapidly exploring random trees [8]. The genetic algorithm will determine the polynomial coefficients which result in a curve with the greatest fitness, as determined by the constraints of the problem (namely path distance and jerk). A polynomial function was selected to describe the path since it inherently results in a continuous trajectory rather than traditional discontinuous waypoint based trajectories.

While there is no absolute best path planning algorithm for all situations, this paper will show that:

- A Genetic Algorithm can effectively plan a path through a robotic manipulator's configuration space.
- The algorithm presented can plan a path which not only minizes distace travelled, but also account for robotic manipulator dynamics such as acceleration and jerk.
- The algorithm can identify a viable solution which is within XXXX% of the shortest distance as determined using a deterministic algorithm such as wavefront, but in much less time.

II. Background Review

Path planning is an active area of research due to the complexity of the problem and the need for a robust solution applicable to various environments. Many algorithms exist, but none of the proposed algorithms are capable of encompassing all the problem constraints for various environments [9].

Existing path planning solutions include reactive motion planning algorithms, graph and probabilistic based motion planning, and optimization based planning [10]. Each of these has advantages and disadvantages with regards to planning a route through a robot's configuration space. Reactive planning algorithms such as potential field and wavefront methods are simple in concept however these are local approach methods and do not consider dynamic constraints. Some are inherently susceptible to local minima, making them unacceptable for obstacle avoidance [11]. The wavefront method is designed to find the shortest path, and as such will be used as the benchmark against which results from the suggested algorithm will be measured. It is expected that while the suggested algorithm will generate a longer path, the dynamic constraints will be satisfied unlike in the wavefront solution.

Graph based planning offers a global approach. Graphs may be generated both deterministically (visibility graphs, cell decomposition, Voronoi diagrams), or randomly (probabilistic roadmaps). [10] Established algorithms (such as Dijkstra's shortest path search) are used to extract the shortest path from the graph [12]. However dynamics of the robot are ignored in the generation of the graph, and this is detrimental to the criterion of path smoothness. Challenges also arise when there are restrictive obstacles such as narrow passages, however research in areas such as probabilistic roadmaps (PRMs) offers promising solutions [13]. Optimization based planning offers more confidence in finding a global optimum, however it is hindered by a lack of robustness: a poorly formulated problem may not converge [14]. It is also difficult to define obstacles.

In an effort to provide a simple method of avoiding obstacles and defining a path, the manipulator's configuration space will be discretized using a grid-based approach, and curve fitting techniques will be employed to find the ideal path. Defining the environment with a grid-graph may not be the most effective (cell decomposition offers greater efficiency [15]), however it provides a simple way to defined obstacles as well as flexibility in representation. The grid will represent the configuration space for an articulated industrial robot. The grid cell size will define the computational resources required. A tradeoff between resolution in the grid and computational efficiency must be made.

To define the fitted curve, research has been done using Bzier curves [16]. Bzier curves provide smooth paths that guarantee obstacle avoidance. While very effective for lower-dimensional problems, quartic or quintic functions begin to get very computationally expensive. That is why this paper aims to use curve fitting in combination with genetic algorithms to optimize the curve. The curve will be optimized by minimizing distance, maximizing smoothness (minimizing jerk), and ensuring a safe passage around all obstacles. Genetic algorithms are ideal for such complex optimization problems and offer good robustness as well [17].

The father of genetic algorithms, John Holland, modeled this machine intelligence technique after the evolution research by Darwin and the genetic "survival of the fittest" discovered in the natural, biological sphere. There are two natural learning epitomes available: the brain and evolution. Holland was inspired by the fact that the processes of natural evolution and natural genetics have become elucidated over decades of study in biology and molecular biology [18]. The subtleties of the fundamental mechanisms of the brain, in contrast, are still shrouded in mystery. As such it seems clear to use the better understood model of genetic evolution as a platform for an optimization technique.

Research has already been done regarding the application of the parallel heuristic search method of genetic algorithms to curve fitting [19]–[21]. The goal is to find the coefficients of a polynomial function that will define a path that with optimized path criteria. Thus each chromosome will contain a set of function parameters, labelled $\{P_4, \ldots, P_0\}$ in Equation 1.

$$\phi = P_4 * \theta^4 + P_3 * \theta^3 + P_2 * \theta^2 + P_1 * \theta + P_0 \qquad (1)$$

Testing will be performed by creating random environments containing arbitrarily place objects. The algorithm will then plot a trajectory between a randomly defined start and end point. Many iterations of these tests will be performed in a variety of environments in order to accurately determine the effectiveness of this algorithm. The average path distance for each environment will be compared to the minimum distance as calculated by a Wavefront algorithm (CITE).

The wavefront algorithm is a deterministic algorithm which is capable of determining the shortest path between two points through a descritized environment. This will provide a "gold standard" which can be used to evaluate the performance of the genetical algorithm. Since the genetic algorithm is a stochastic search, it will never result in an identical path twice. As such all results will be presented as averages and their associated distributions.

The result will be a global path planning algorithm functional in a static environment. Dynamic environments are possible with the proposed algorithm, however computational efficiency must be optimized in order to minimize the time required to find the solution. This new applications offers a simple way to determine a path while taking into consideration all the path criteria.

III. Method

The Genetic Algorithm (GA) formed the basis for the experiemnts. Significant effort was expended setting up the GA in advance of testing, since without a well structured algorithm, the testing was sure to show poor results. The GA was implemented on Matlab, using the built-in GA libraries in order to avoid unnecessary development. While the base functionality was used, there remained many design considerations.

A. Criteria

B. Configuration Space

Commanding a robotic manipulator's position is complicated by the fact that it cannot be specified is X-Y-Z coordinates, but rather must be commanded using joint angles. Not only is this not intuitive for a human, but existing planning techniques function using more standard X-Y or X-Y-Z coordinate systems. As such, the robot's workspace was converted into a Configuration Space. Figure 1 shows both the workspace and the configuration space for a 2-DOF robot in an arbitrary workspace. The robotic manipulator's two degrees of freedom are denoted by the blue and red lines (each DOF respectively). The alloable robot configurations denoted in the Configuration Space by whitespace, while collisions are denoted by black.

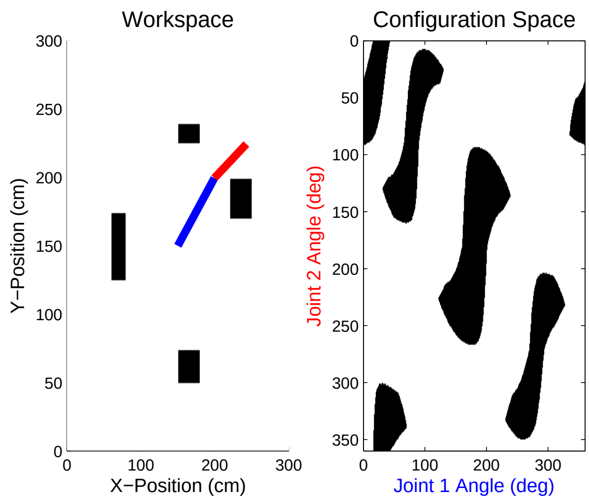

Fig. 1. Conversion from robot workspace to configuration space

In the configuration space, robot tracjectories can be easily identified by either a human or a motion planning algorithm. This angle-angle plot can then be used to implement a polynomial trajectory, of which optimal coefficients can easily be solved using a Genetic Algorithm. The parameterized trajectory is denoted in Equation 1.

C. Population Initiatlization

The population was randomly initialized with coefficient values in the range of 100 to 100. This range was imposed on the coefficients as it was found to provide a good compromise between promoting function manipulation while at the same time keeping it under control so that changes to coefficients did not make too drastic a change to the shape of the path.

The population size was chosen to be 125. This population number was found to provide the necissary diversity in the population.

D. Encoding

The GA was developed using a linear encoding scheme, where a chromosome was defined to be an array of the polynomial function's coefficients (denoted as $\{P_0, \ldots, P_4\}$ in Equation 1). Therefore, the GA would ultimately return the ideal set of coefficients for a function which would describe the optimal path for the robotic manipulator. These coefficients were limited to a range of [-50,50], since larger coefficients guarantee an invalid path since it would be too large and leave the workspace. While larger coefficients do not adversely affect the result of the algorithm, the increased diversity requires significantly larger computational resources which were not avaliable.

E. Selection

A tournament selection method was used in this implementation. 2 individuals were chosen for each tournament. This allowed for an acceptable tradeoff between choosing a fit parent while promoting diversity in the offspring. This was one approach used to increase the diversity of the population when it was observed that the algorithm was getting stuck in local minima.

F. Reproduction and Mutation

A heuristic reproduction method was used wherein the child is generated by taking the mean of the coefficients of the two parents, and then biased to resemble the parent. In our implementation, the child was biased 20% towards the most fit parent using formula 2.

$$child_i = parent_{i2} + 1.2 * (parent_{i1} + parent_{i2}) \tag{2}$$

As is common in non-binary encoded GA problems, this crossover method is based on a mean of the parents' genes. This is a key distinction from a traditional point crossover method, as it allows the genes to evolve from values that were not present in the initial population. If point crossover methods had been used, the best case scenario would involve the optimal configuration of coefficients from the initial, random population.

Mutation was achieved using a completely random mutation to a coefficient within the inidvidual to be mutated. In order to satisfy the linear constraints of the start and end points, the function was rendered invalid by the mutation, a new random coefficient was chosen instead.

G. Termination

KARL

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

IV. Results

V. Discussion

References

[1] A. Elshamli, H. A. Abdullah, and S. Areibi, "Genetic algorithm for dynamic path planning," *Canadian Conference on Electrical and Computer Engineering, IEEE*, vol. 2, pp. 667–680, 2004.

[2] L. Kavraki and J. C. O. M. Svestka, P.AND Latombe, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 556–580, 1996.

[3] M. Sharir, "Algorithmic motion planning in robotics," *Computer*, vol. 22, no. 3, pp. 9–19, 1989.

[4] R. Khosla and P. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," *IEEE International Conference on Robotics and Automation, Proceedings*, pp. 1778–1784, 1988.

[5] E. Rimon and D. Doditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[6] S. Yang and M. Meng, "An efficient neural network approach to dynamic robot motion planning," *Neural Networks*, vol. 13, no. 2, pp. 143–148, 2000.

[7] G. Renner and A. Ekart, "Genetic algorithms in computer-aided design," *Computer-Aided Design*, vol. 35, no. 8, pp. 709–726, 2003.

[8] S. Rodriguez, T. Xinyu, L. Jyh-Ming, and N. Amato, "An obstacle-based rapidly-exploring random tree," *IEEE International Conference on Robotics and Automation, Proceedings*, pp. 15–19, 2006.

[9] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," *IEEE 4th Student Conference on Research and Development, SCORed*, pp. 183–188, 2006.

[10] S. Waslander, *ME597: Autonomous Mobile Robotics - Section 8: Planning I (Course notes)*. University of Waterloo, 2012.

[11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *IEEE International Conference on Robotics and Automation, Proceedings*, pp. 1398–1404, 1991.

[12] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 20, no. 4, pp. 269–271, 1959.

[13] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planner," *IEEE International Conference on Robotics and Automation, Proceedings*, vol. 3, pp. 4420–4426, 2003.

[14] S. Waslander, *ME597: Autonomous Mobile Robotics - Section 8: Planning III (Course notes)*. University of Waterloo, 2012.

[15] F. Lingelbach, "Path planning using probabilistic cell decomposition," *IEEE International Conference on Robotics and Automation, Proceedings*, vol. 1, pp. 467–472, 2004.

[16] J. Choi, R. E. Curry, and G.H., "Real-time obstacle-avoiding path planning for mobile robots," *Proceedings of the AIAA Guidance, Navigation and control Conference*, 2010.

[17] X. Zou, B. Ge, and P. Sun, "Improved genetic algorithm for dynamic path planning," *International Journal of Information and Computer Science*, vol. 1, no. 2, 2012.

[18] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2, pp. 95–99, 1988.

[19] M. Gulsen, A. E. Smith, and D. M. Tate, "A genetic algorithm approach to curve fitting," *International Journal of Production Research*, vol. 33, no. 7, pp. 1911–1923, 1995.

[20] C. L. Karr, D. A. Stanley, and B. J. Scheiner, *Genetic Algorithm Applied to Least Squares Curve Fitting*. US Department of the Interior, Bureau of Mines, 1991.

[21] A. T. Ismail, A. Sheta, and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment," *Journal of Computer Science*, vol. 4, no. 4, pp. 341–344, 2008.